GEO: A GEOGRAPHIC DATABASE SYSTEM FOR REMOTE SENSING

Tao Chen and jian-kang Wu
Image Processing Lab.
Dept.of Radio Electronics
Univ. of Science and Technology of China
Hefei, China
Commission IV

ABSTRACT

Database system is a very important part for remote sensing technique. This paper presents the design of a knowledge-based geographic database system GEO for remote sensing. In GEO, geographic data(remote sensed images and their processing results, elemental maps, some thematic maps) are organized and stored using object-oriented run-length coding. The database model of GEO is an extended relational model allowing for frame-like structure(abstract data type ADT). The ADT can be an ordinary data type, such as character string, integer, real number, it can also be procedure name, frame-like structure, which represents knowledge of geographic data, ecological and domain specific knowledge. Different user views of the content and organization of geographic data are supported in GEO. Query processing in GEO is knowledge-guided, therefore it is much quicker than usual. The query items may include components of ADT, computer-aided planing and decision making can be implemented without much difficulty. The extended DBMS of GEO can deal intelligently with complex geographic data and their interrelations.

This project is supported by China Science Research Fund under the contract No.84TECH002

1. INTRODUCTION

A database model(data model) specifys the structure of a database and the related operations which may be performed on that database. This specification is called a schema. Many methods have been attempted to extend the relational model to contain semantic and other relevant knowledge[1,2] in order to provide richer, more expressive concepts with which to capture more meaning than using classical data models. Attributes' columns of a relation are very simple, attributes can be extended to act as roles by appropriate naming and by adding role oriented operations. ADT has been considered as a central theme in a relational database context [3,4]. In this paper we propose such an extension, columns of a relation as an abstract data type (frame-like structure). We substitute some columns of a relation with ADTs, attemtting to add knowledge about geographic data and applicational fields to relations in GEO, and using attached procedures to give semantic interpretation of geographic data and to aid queries.

In what follows we will first give a brief description of GEO system [5] and the extended database scheme, we then introduce knowledge-quided query language KGQL, and illustrate its usage for special queries and decision-making.

2.GEO SYSTEM ORGANIZATION

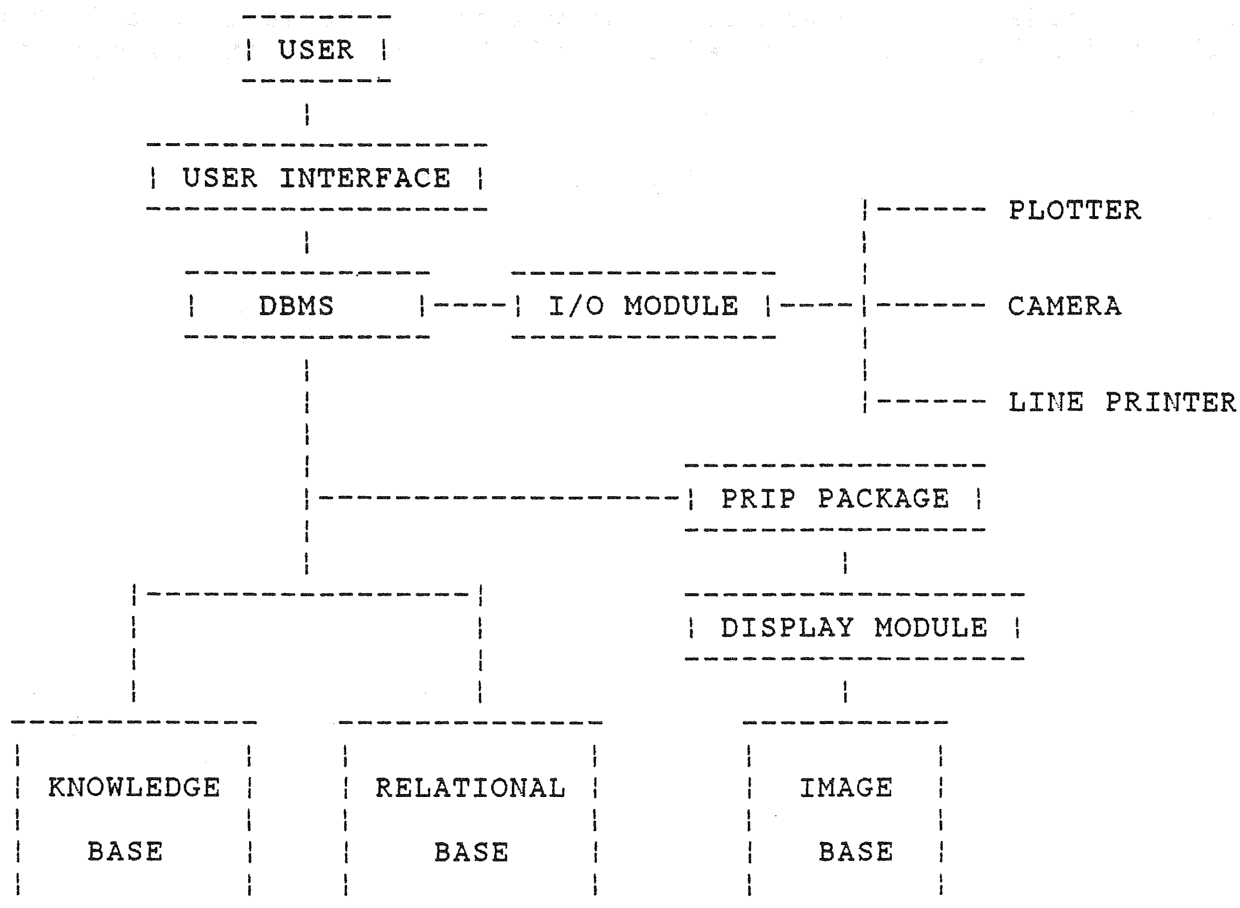

Fig 1. The simple diagram of GEO system

A prototype system of GEO has been designed and implemented at Image Processing Lab. of Uniersity of Science and Technology of China, on an image processing system of VAX11/750 + Model75 image processor. GEO employs an extended relational model allowing ADTs, which use frames proposed by Minsky in [6] to represent the spatial relationships, ecological knowledge, and other relevant knowledge of image entities. The diagram of GEO is shown as Figure 1.

GEO is object-oriented, it structures information around the objects of the domain and to organise objects in hierarchies through which inheritance of properties takes place. There are three types of data managed by GEO system:

A. original images -- the remote sensed images. They provide an instrict views of data.

B. relations -- the processing results of original images and various maps. They, stored in various relations, are digital forms of geographic elemental maps (such as river map, road map, residential map, elevation map, district map, etc.), some thematic maps (such as land cover map, forest map, etc.), and topological maps.

C. descriptions -- the frame subsystem in integrated knowledge base. It stores attributes and knowledges (attributes of objects, spatial relationships between objects, influences of one object over another, etc.). This kind of knowledge is readily available and well-structured. Reasoning in GEO is very convenient because all spatial data and knowledge are integrated and managed by DBMS of the system.

3.EXTENDED DATABASE SCHEME

The principle shortcoming of current database systems is the descriptives of their underlying data models.

The best known classical data model are the relational, the hierarchical, and network models. All the three models are based on the record's model, which arranges data in fixed linear sequences of field values. Records are machine oriented--they provide an efficient basis for storing and processing data but are not user oriented at all. Take the relational model, which is considered the most user oriented among the three classical database models, as an example, the main attraction of the relational model is its mathematical clarity, which facilitates the formulation of nonprocedural high-level queries and thus seperates the user from the internal organization of data, but it is far from being able to satisfy the requirements of an increasing diversified user community. Perhaps the primary attraction of relational systems is that their tabular model is easily understood, however the simplicity of the model does have its drawbacks. The most significant in term of application description is that all relationships between objects must be based on data values. This limitation forces the scheme designer to create dummy fields or additional relations to describe logical associations between data elements. The only nonatomic entities in the classical database are records, the attributes of records are necessarily atomic strings, integers, reals, or Booleans.

GEO's changed the structure and content of database model so that the interpretation of data interrelationships and frame-based problem solving can take place as parts of the database management system. Using the model, we can provide a higher level model and data abstraction capability, support more than one sort of data relationships, and provide more complex constraints for specifying databases than record-oriented models. The essential improvement in the semantic model on GEO is that the individual attributes in a record can be nonatomic themselves.

GEO's database schema adopts an extented knowledge-based relational data model, geographic entity is represented by four representations, namely, its logical, physical, frame and the corresponding iconic representations. The logical representation of image entity is the data dictionary, the physical representation the geometric coordinate data, and the frame subsystem are the semantic interpretation of geographic data. The compoments of a relation in GEO are classical attributes and frames. The iconic representations of geographic entities are hierarchical organized micro-images of the entities. They give visual views of geographic entities and aim the visual queries of GEO.

Some basic relations of GEO are shown as followings:

ROADNAME(PID,NAME,ROID,RI-ADT)
CITYNAME(PID,NAME,CIID,CI-ADT)
ROADS(PID,ROID,X,Y,LENGTH)
CITIES(PID,CIID,X,Y,LENGTH)

The relations RIVERS and CITIES are defined on the same domains specifying the image identification, entity identification code, X-coordinate and Y-coordinate of start point of line segments, and the length of the line segments(object oriented run-length coding). The names of rivers and cities are defined in the extended relations RIVERNAME and CITYNAME. RIVERNAME and CITYNAME are abstract data types (ADTs). The fields CI-ADT, RI-ADT are data types, which are represented by AI specified frames. We associate a 'frame' ADT with a relation called CITYNAME, which has attributes NAME, PID, CIID. The frame of a city consists of many slots, each slot contains a kind of feature attributes of the city. The relationships between RIVERNAME, CITYNAME and RIVERS, CITIES are associated by ASSOCIATE command explicitly.

4.FRAME AS AN ADT

Knowledge consists of the symbolic descriptions, which characterize the definitional and empirical relationships in a domain, and the procedures for manipulating these descriptions. There are a number of ways that semantic knowledge can be logically represented, these include production rules, semantic nets, frame, and mathematic logic[7,8,9,10]. In GEO, we represent knowledge using frames in AI. A frame is a complex data structure for representing a stereotypical situation. The frame has slots for the objects that play a role in the stereotypical situation as well as relations between these slots. Attached to each frame are different kinds of information, such as how to use it, what to do if something unexpected happened, default values for its slots, etc.

The structure of a possible frame REGION in GEO would be:

```
|--------------------------------------------------------------|
| REGION                                                       |
|       IS-A:    RANGE ( entity )                              |
|       NAME:    RANGE ( c10 )                                 |
|                IF-ADDED ( print the name on the terminal )   |
|                IF-NEEDED ( find the value from the           |
|                           field NAME of the data dictionary ) |
|       EXAMPLES: ( city1... )                                 |
|       PARENT: RANGE ( c10 )                                  |
|       CENTER: DEFAULT ( integer-range 40 1000 )              |
|                IF-NEEDED ( find the coordinate of the center |
|                            point using procedure CENT-R )    |
|       CLASS:   DEFAULT ( i4 )                                |
|                IF-NEEDED ( call procedure GET-CLASS )        |
|       AREA:    DEFAULT ( i4 )                                |
|                IF-NEEDED ( call procedure AREA-R )           |
|       PERIMETER:DEFAULT ( integer-range 15 600 )             |
|                IF-NEEDED ( get the value by calling          |
|                           procedure LENGTH )                 |
|       ICON:    RANGE ( c10 )                                 |
|                IF-ADDED ( show the icon on display screen    |
|                            by procedure DISPLAY )            |
|                IF-NEEDED ( call procedure GET-NAME )         |
|       POPULATION:DEFAULT ( integer-range 1 19999 )           |
|                IF-NEEDED ( get the value from the table or   |
|                           inquire the database adminstrator )|
|       THROUGH:RANGE ( river, road, rail )                    |
|                IF-NEEDED ( find the name list of rivers,     |
|                            roads and rails which pass through|
|                            the region )                      |
|                IF-ADDED ( print the list, display the        |
|                            entities on display screen )      |
|       PLANTING:RANGE ( list and distribution icon )          |
|                IF-NEEDED ( find out the place suitable for   |
|                           planting a given kind of trees )   |
|                IF-ADDED ( print the list, display the        |
|                            distribution icon )               |
|       TRANSPORTATION:RANGE( i4 )                             |
|                IF-NEEDED ( find the total transportation     |
|                            capacity of all rivers, roads     |
|                            and rails through the city        |
|                IF-ADDED ( print the sum )                    |
|--------------------------------------------------------------|
```

Where, c10 is a character string of 10 bytes, i4 represents an integer number of 4 bytes, IF-NEEDED is a procedure which performs the filling of a slot, and IF-ADDED is a matching procedure which is triggered by the slot data. The IS-A and PARENT slots indicate the hierarchical relationship of entities, the entity can inherit the properties of its parent.

A user might define an CITYNAME relation as follows:

```
create  CITYNAME ( NAME = c10,
                   PID = i4,
                   CIID = i4,
                   CI ADT = region)
```

Where 'region' is a frame with the definition shown above.

Tuples can be added to relation CITYNAME using following command:

```
append  CITYNAME ( NAME = "city1",
                   PID = 12,
                   CIID = 56,
                   CI ADT = *)
```

The first three fields can be converted and stored correctly, except the symbol '*'. GEO fills the slots of the frame automatically by using image algebra operations defined in IF-NEEDED procedures.

5. Knowledge guided query language KGQL

The query language KGQL provides a versatile interface to users. All operatons are displayed on screen in menus snd forms. A user get his/her query done by the selection of menus and the filling of the forms.

KGQL, on the basis of image algebra and ADTs, extends the query items belonging to several different relations and being relations and slots in frames, so that almost all kinds of queries can be performed. KGQL can do the conventional queries as well as the region retrievals quickly, perform the topological retrievals based on the natural features or structures of image, similarity retrievals, and visual queries. KGQL has been extended to support abstraction, procedures and more complex objects. KGQL uses its knowledge about goals, constraints, and conflict resolutions to ask the database user appropriate questions, which are structured dynamically in response to data returned by the database manager.

We have defined normal comparison, arithmetic and aggregate operations for the new data type. A user could use stardard operators on the frame domain:

CIID	NAME	ID	...
	.P		

IS-A	
...	
AREA	> 210
...	

GEO matchs the query item with field names of the relation CITYNAME, if successfully, do the conventional query. If fail, in this case, GEO matchs it with the slot names of field FRAME, and use the value of the slot matched as a conventional attribute value to get the query done.

GEO provides aggregate functions for the FRAME column. For example, a user can find all the cities which PID code is 75 and perimeter is large than 110KM in the nearest meaning as follows:

PID	CNAME	...
.P	75	

IS-A	
...	
PERIMETER	MIN > 110
...	

The select operators of KGQL also include arithmetic operators. One can perform the topological query such as "find the city which center point is below the line 3X+8Y=600" using the fllowing command:

PID	CNAME	...
.P	.P	

IS-A	
...	
CENTER	3*X+8*Y < 600
...	

A user can also use KGQL to find the rivers, rails and roads which pass through a city, said city1(PID = 12):

PID	CNAME	...
12	city1	

IS-A	
...	
THROUGH	* .DIS

Where '*' indicates the triggering of the slot THROUGH.

GEO call the corresponding frame of city1, triggers the slot THROUGH to get the name list of rivers, roads and rails which pass through city1 and display these entities in display screen on request.

Finally, the query language KGQL can aid some sort of decision making. For instance, in order to decide wether to build a new railway in city1 or not, we must find out the total transportation capacity of city1 in advance to see if it is necessary, we can using the following command to do such job:

PID	CNAME	...
12	city1	

IS-A	
...	
TRANSPORTATION	* .DIS

6. CONCLUSIONS

We have proposed a knowledge based geographic database system GEO, and given a brief description of the system organization. We used an extended relational schema with ADTs, so that the query items of KGQL might include components of ADTs and the query processing was guided by knowledge. Moreover, GEO provides powerful reasonings and decision-making functions. We hope that further extention of GEO's data model can lead compatible and efficient modules such as model base and decision-support system, and the DBMS of GEO can deal much more intelligently with complex image entities and relationships.

REFERENCES

1. M.L.Brodie, J.Mylopoulos, and J.W.Schmidt(Ed.), On Conceptual Modeling, Springer-Verlag, 1984.

2. J.W.Schmidt, "Type Concepts for Database Definition", Proc. International Conference on Data Bases, Haifa, Israel, August 1978.

3. L.A.Rowe, and K.A.Schoens, Data Abstraction, Views and Update in RIGEL, Proc. 1979 ACM SIGMOD International Conference on the Management of Data, Boston, Mass., May 1979, pp.71-81.

4. L.Bic, J.P.Gilbert, Learning from AI: New trends in database technology, Computer, March 1986.

5. Tao Chen, GBASE---A Knowledge-Based geographic Image Database, Master desertation, University of Science and Technology of China, January 1987.

6. M.Minsky, A framework for representing knowledge, in P.Winston (Ed.) The psychology of computer vision, McGraw-Hill, p211-277,1975.

7. P.H.Winston, Artificial Intelligence, second edition, Addison-Wesley Publishing Company, Inc. 1984.

8. P.R.Cohen and E.A.Feigenbaum (Ed.), The Handbook of Artificial Intelligence, Volume 1, 2, and 3, Addison-Wesley Publishing Company, Inc. 1984.

9. Frederick Hayes-Roth, Donald A. Waterman, and Douglas B.Lenat, Building Expert Systems, Addison-Wesley Publishing Company, Inc. 1983.

10. D.J.Hartzband, F.J.Maryanski, Enhancing knowledge representation in Engineering Databases, Computer, Sept. 1985.